

## Course 10

# 削減與搜尋

Prune and Search

# ■ Outlines

---

## ◆ 本章重點

- Prune and Search概念
- A Simple Example: 二分搜尋法
- 挑選第 $k$ 小的元素 ( $k^{\text{th}}$  Selection) 問題

## ■ Prune and Search概念

- ◆ 可視為切割與征服 (Divide and Conquer) 策略的特例。
  - 在一個資料集合當中，依照目前問題的特性，**事先去除掉一些答案不可能存在的資料子集**，再由目前剩下的資料子集繼續嘗試搜尋答案。
  - 包含多次的處理，每次的處理都會將資料集合**刪除固定的百分比**，並運用同樣的方法**遞迴地**以刪除後的資料當作輸入資料重新解問題。經過若干次處理後，資料量將可縮小到能用**固定常數的時間**解得答案。
- ◆ 它的精神所在便是如何**有效地刪減資料集合**，就像切割與征服 (Divide and Conquer) 策略強調的如何有效的合併。不過由於利用削減搜尋法得出來的演算法，有時會和分割解決法所得的結果相同，因此這兩種方法時常被混淆。

## ■ A simple example: 二分搜尋法

- ◆ 有一排序後之序列如下 : (Search 9)

~~1~~   ~~4~~   ~~5~~        7        9        10        ~~12~~   ~~15~~

step 1



step 2



step 3



- ◆ 在每一次比較以後，總是會有一群資料被削減掉 (**pruned away**).
- ◆ Binary search 可視為特殊的切割與征服策略!!這是因為：
  - 切割與征服策略強調將母問題切割成較小的問題 (**Divide**)，再使用相同的解決程序對所有小問題加以分別處理 (**Conquer**)。所有小問題的解可以成為母問題的最後解; 若有必要，則再將每個小問題的處理結果加以合併。
  - 但二分搜尋法將問題切割及經過一次比較後，會直接將答案不存在之一方的所有資料削減掉 (**Prune**)，再使用相同的程序去搜尋另一方 (**Search**)。

## ■ 挑選第 $k$ 小的元素問題 ( $k^{\text{th}}$ Selection)

---

### ◆ 問題定義：

- 已知有 $n$ 個元素的集合 $S$ 。求一個Algorithm可找出 $S$ 中第 $k$ 小的數。
- 特別要求：在 $O(n)$ 內找出解答。

### ◆ 此問題的直觀解法：

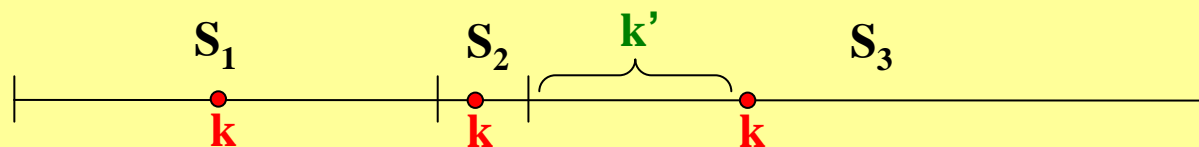
- 先對 $n$ 個元素排序
- 再由小到大依序找出第 $k$ 個數值即為解答

### ◆ 直觀解法的時間複雜度： $O(n \log_2 n)$

### ◆ 採Prune and Search解法的時間複雜度： $O(n)$

- ◆ 令集合  $S = \{a_1, a_2, \dots, a_n\}$ ，挑出一個元素  $p \in S$ ，利用  $p$  將  $S$  分成三部份  $S_1, S_2, S_3$ :
  - $S_1 = \{a_i \mid a_i < p, 1 \leq i \leq n\}$  // 其內元素皆小於  $p$
  - $S_2 = \{a_i \mid a_i = p, 1 \leq i \leq n\}$  // 其內元素皆等於  $p$
  - $S_3 = \{a_i \mid a_i > p, 1 \leq i \leq n\}$  // 其內元素皆大於  $p$
- ◆ 有三個判斷條件以找出第  $k$  小的元素:
  - If  $|S_1| \geq k$ , then 第  $k$  小的元素存在於  $S_1$ , prune away  $S_2$  and  $S_3$ 。
  - else, if  $|S_1| + |S_2| \geq k$ , then  $p$  即為第  $k$  小的元素。
  - else, 第  $k$  小的元素存在於  $S_3$  中, prune away  $S_2$  and  $S_3$ 。令  $k' = (k - |S_1| - |S_2|)$ , 在  $S_3$  中找第  $k'$  個元素即為解答。

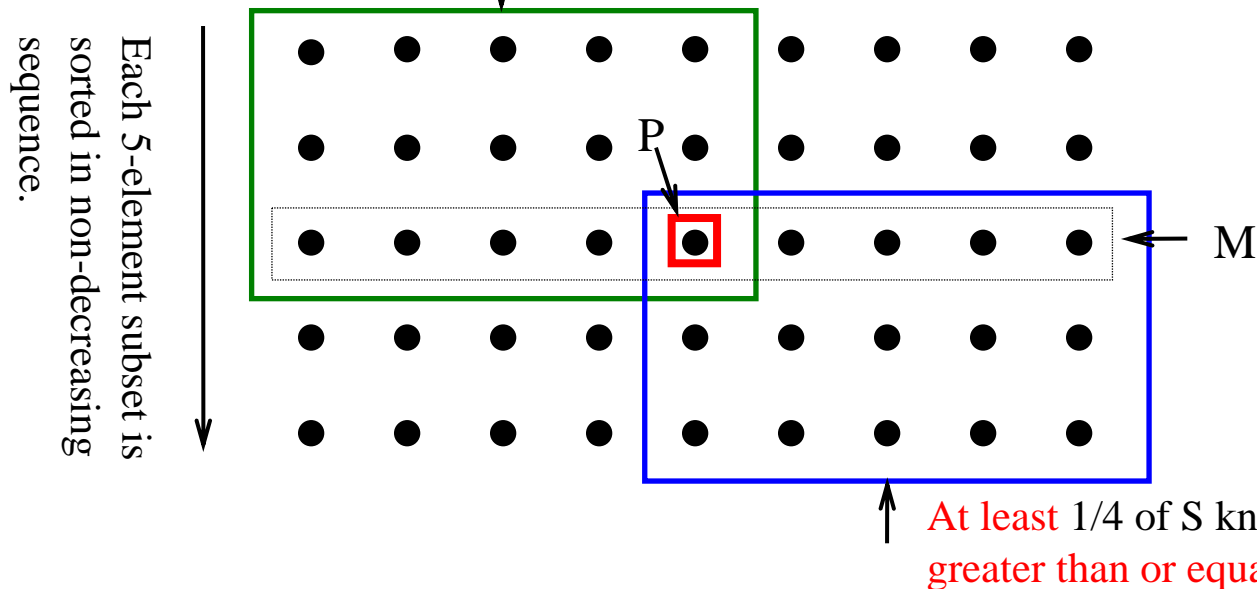
說明圖示：



## ◆ 如何挑選p?

- 將n個元素區分成5個一組，分行存放。若最後一組元素不足5個，則以 $\infty$ 補足
- 把每一組由小到大排序，每組第三大元素即是組中位數M
- 選p為所有組中位數 (有  $\lceil \frac{n}{5} \rceil$  個) 的中位數 (by recursive call)

At least 1/4 of S known to  
be less than or equal to P.



# 演算法

◆ Input: A set  $S$  of  $n$  elements.

◆ Output: The  $k$ th smallest element of  $S$ .

- Step 1: 將  $S$  分成  $\lceil n/5 \rceil$  組資料集合，每一組有5個資料，不足5個資料以  $\infty$  補足。
- Step 2: 排序每一組資料
- Step 3: 找出所有組中位數的中位數
- Step 4: 將  $S$  區分成三部份  $S_1$ ,  $S_2$  and  $S_3$ , which contain the elements less than, equal to, and greater than  $p$ , respectively.
- Step 5: 利用三個判斷條件以找出第  $k$  小的元素:
  - If  $|S_1| \geq k$ , then 第  $k$  小的元素存在於  $S_1$ , prune away  $S_2$  and  $S_3$ 。
  - else, if  $|S_1| + |S_2| \geq k$ , then  $p$  即為第  $k$  小的元素。
  - else, 第  $k$  小的元素存在於  $S_3$  中, prune away  $S_1$  and  $S_2$ 。令  $k' = (k - |S_1| - |S_2|)$ , 在  $S_3$  中找第  $k'$  個元素即為解答。



## 範例

◆  $S = \{1, 25, 2, 24, 3, 23, 4, 22, 5, 21, 6, 20, 7, 19, 8, 18, 9, 17, 10, 16, 11, 15, 12, 14, 13\}$ ，找第 $k$ 小的元素。

Ans

[Step 1]

1	23	6	18	11
25	4	20	9	15
2	22	7	17	12
24	5	19	10	14
3	21	8	16	13

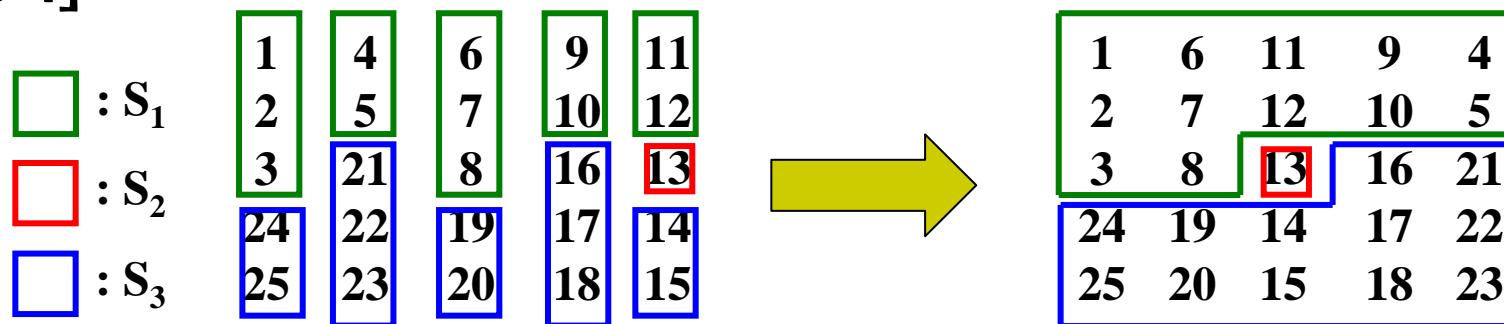
[Step 2]

1	4	6	9	11
2	5	7	10	12
3	21	8	16	13
24	22	19	17	14
25	23	20	18	15

[Step 3]

1	4	6	9	11
2	5	7	10	12
3	21	8	16	13
24	22	19	17	14
25	23	20	18	15

[Step 4]

[Step 5] 利用三個判斷條件以找出第 $k$ 小的元素

- 若  $k = 11$  (搜尋範圍  $|S_1|$ )
- 若  $k = 13$  (搜尋範圍  $|S_1| + |S_2|$ )
- 若  $k = 22$  (搜尋範圍  $|S_3|$ )

## Time complexity

---

- ◆ At least  $n/4$  elements are pruned away.
- ◆ The problem remaining in step 5 contains at most  $3n/4$  elements.
- ◆ Time complexity:  $T(n) = O(n)$ 
  - step 1:  $O(n)$  //掃一輪即可得知
  - step 2:  $O(n)$  //有 $\lceil n/5 \rceil$ 組資料，每組資料排序需固定常數時間 $O(1)$
  - step 3:  $T(n/5)$  //採遞迴方式找尋，共有 $\lceil n/5 \rceil$ 組
  - step 4:  $O(n)$  //掃一輪即可得知
  - step 5:  $T(3n/4)$  //每次Prune掉至少 $n/4$ 資料量後，尚有 $3n/4$ 左右的剩餘資料需遞迴執行
- ◆ 遞迴方程式為 $T(n) = T(3n/4) + T(n/5) + O(n)$ ，採遞迴樹法分析，可得知此演算法的時間複雜度為  $O(n)$